

 [Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search:  The ACM Digital Library  The Guide

reverse engineering

THE ACM DIGITAL LIBRARY

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used reverse engineering

Found 61,162 of 184,245

Sort results by relevance

 Save results to a Binder[Try an Advanced Search](#)

Display results expanded form

 [Search Tips](#)Try this search in [The ACM Guide](#) Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale **1 Article abstracts with full text online: Why teach reverse engineering?** Muhammad Raza AliJuly 2005 **ACM SIGSOFT Software Engineering Notes**, Volume 30 Issue 4**Publisher:** ACM PressFull text available:  [pdf\(206.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software reverse engineering is a fascinating discipline of software engineering. But it has failed to attract attention from students. Largely due to the facts that many universities around the world do not offer relevant courses, developing new software has always been considered superior than to maintain existing systems. But owing to the arrival of the internet, and client-server technology. Many organizations wish to adapt their existing systems. Thus the trend has somewhat shifted towards ...

**Keywords:** internet, legacy systems, obfuscate, plagiarism, program understanding, reengineering, software engineering, software evolution and maintenance, software reverse engineering

**2 Design technologies: A software reverse engineering experience**

Erich Buss, John Henshaw

October 1991 **Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research****Publisher:** IBM PressFull text available:  [pdf\(1.19 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper reviews the progress-to-date of the application of program reverse engineering technologies to a large-scale legacy software product. Basic reverse engineering concepts and a project overview are outlined, followed by a description of the legacy software product, the reverse engineering toolkit used, and analysis and discussion of the experiences so far. Future research directions and summary comments are then detailed.

**3 Evaluating the reverse engineering capabilities of Web tools for understanding site content and structure: a case study**

Scott Tilley, Shihong Huang

July 2001 **Proceedings of the 23rd International Conference on Software Engineering****Publisher:** IEEE Computer SocietyFull text available:  [pdf\(360.47 KB\)](#) Additional Information:

[Publisher Site](#)[full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

*This paper describes an evaluation of the reverse engineering capabilities of three Web tools for understanding site content and structure. The evaluation is based on partitioning Web sites into three classes (static, interactive, and dynamic), and is structured using an existing reverse engineering environment framework (REEF). This case study also represents an initial evaluation of the applicability of the REEF in the related but qualitatively different domain of Web sites. The case stu ...*

#### 4 Reverse engineering: a roadmap

Hausi A. Müller, Jens H. Jahnke, Dennis B. Smith, Margaret-Anne Storey, Scott R. Tilley, Kenny Wong

May 2000 **Proceedings of the Conference on The Future of Software Engineering**

Publisher: ACM Press

Full text available: [pdf\(1.53 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



**Keywords:** data reverse engineering, program comprehension, program understanding, reverse engineering, software analysis, software engineering, software evolution, software maintenance, software migration, software reengineering, software tools, tool adoption, tool evaluation

#### 5 Reverse engineering and system renovation—an annotated bibliography

M. G. J. van den Brand, P. Klint, C. Verhoef

January 1997 **ACM SIGSOFT Software Engineering Notes**, Volume 22 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.32 MB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)



To facilitate research in the field of reverse engineering and system renovation we have compiled an annotated bibliography. We put the contributions not only in alphabetical order but also grouped by topic so that readers focusing on a certain topic can read their annotations in the alphabetical listing. We also compiled an annotated list of pointers to information about reverse engineering and system renovation that can be reached via Internet. For the sake of ease we also incorporated a brief ...

**Keywords:** annotated bibliography, reverse engineering, system renovation

#### 6 Reverse engineering of legacy code exposed

Bruce W. Weide, Wayne D. Heym, Joseph E. Hollingsworth

April 1995 **Proceedings of the 17th international conference on Software engineering**

Publisher: ACM Press

Full text available: [pdf\(599.73 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



#### 7 Reveng: a cost-effective approach to reverse-engineering

Xavier A. Debest, Rüdiger Knoop, Jürgen Wagner

October 1992 **ACM SIGSOFT Software Engineering Notes**, Volume 17 Issue 4

Publisher: ACM Press

Full text available: [pdf\(755.22 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)



Like many other Software Engineering buzzwords, "Reverse Engineering" emerges as an attempt to formalize some specific tasks, which have long been performed by almost

every software development or maintenance team, but which were not judged worth before to be considered an engineering discipline. The emergence of powerful Computer Aided Software Engineering (CASE) tools opens new opportunities for the cost-effective re-use of the investments made in existing software systems, and thus for the de ...

## 8 A reverse engineering environment based on spatial and visual software

### interconnection models

H. A. Müller, S. R. Tilley, M. A. Orgun, B. D. Corrie, N. H. Madhavji

November 1992 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fifth ACM SIGSOFT symposium on Software development environments SDE 5**, Volume 17 Issue 5

Publisher: ACM Press

Full text available:  pdf(1.28 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Reverse engineering is the process of extracting system abstractions and design information out of existing software systems. This information can then be used for subsequent development, maintenance, re-engineering, or reuse purposes. This process involves the identification of software artifacts in a particular subject system, and the aggregation of these artifacts to form more abstract system representations. This paper describes a reverse engineering environment which uses the s ...

## 9 Reverse engineering: Management decision support through reverse engineering technology

Scott R. Tilley

November 1992 **Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research - Volume 1**

Publisher: IBM Press

Full text available:  pdf(748.41 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Managers of large software systems face enormous challenges when it comes to making informed project-related decisions. They require a high-level understanding of the entire system *and* in-depth information on selected components. Unfortunately, many software systems are so complex and/or old that such information is not readily available. Reverse engineering---the process of extracting system abstractions and design information from existing software systems---can provide some of this mis ...

## 10 Legally speaking: Reverse engineering under siege

### Pamela Samuelson

October 2002 **Communications of the ACM**, Volume 45 Issue 10

Publisher: ACM Press

Full text available:  pdf(84.35 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

 html(41.51 KB)

Is reverse engineering a lawful way to acquire trade secrets?

## 11 Java implementation verification using reverse engineering

David Cooper, Benjamin Khoo, Brian R. von Konsky, Michael Robey

January 2004 **Proceedings of the 27th Australasian conference on Computer science - Volume 26 ACSC '04**

Publisher: Australian Computer Society, Inc.

Full text available:  pdf(549.74 KB)

Additional Information: [full citation](#), [abstract](#), [references](#)

An approach to system verification is described in which design artefacts produced during forward engineering are automatically compared to corresponding artefacts produced during reverse engineering. The goal is to automatically determine if an implementation is consistent with the original design. In the system described, XML Metadata Interchange

(XMI) representations of Unified Modelling Language (UML) class diagrams are recovered from compiled Java class files. These are automatically compar ...

- 12 [Software evolution: Understanding software systems using reverse engineering technology perspectives from the Rigi project](#)   
Hausi A. Müller, Scott R. Tilley, Kenny Wong  
October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1**  
Publisher: IBM Press  
Full text available:  [pdf\(785.90 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Software engineering research has focused mainly on software construction and has neglected software maintenance and evolution. Proposed is a shift in research from synthesis to analysis. Reverse engineering is introduced as a possible solution to program understanding and software analysis. Presented is reverse engineering technology developed as part of the Rigi project. The Rigi approach involves the identification of software artifacts in the subject system and the aggregation of these artif ...

**Keywords:** legacy software, program understanding, reverse engineering, software evolution

- 13 [A specification matching based approach to reverse engineering](#)   
Gerald C. Gannod, Betty H. C. Cheng  
May 1999 **Proceedings of the 21st international conference on Software engineering**  
Publisher: IEEE Computer Society Press  
Full text available:  [pdf\(1.27 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** formal methods, reverse engineering, software maintenance

- 14 [Practical legal aspects of software reverse engineering](#)   
 Brian C. Behrens, Reuven R. Levary  
February 1998 **Communications of the ACM**, Volume 41 Issue 2  
Publisher: ACM Press  
Full text available:  [pdf\(68.69 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

- 15 [Pattern-based reverse-engineering of design components](#)   
Rudolf K. Keller, Reinhard Schauer, Sébastien Robitaille, Patrick Pagé  
May 1999 **Proceedings of the 21st international conference on Software engineering**  
Publisher: IEEE Computer Society Press  
Full text available:  [pdf\(1.43 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** design component, design pattern, design recovery, object-oriented design, reverse-engineering, tool support, visualization

- 16 [Program and interface slicing for reverse engineering](#)   
Jon Beck, David Eichmann

**May 1993 Proceedings of the 15th international conference on Software Engineering****Publisher:** IEEE Computer Society PressFull text available: [pdf\(1.10 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)**17 Model transformation (MT 2006): Application modeling using reverse engineering****techniques**

T. Katsimpa, Y. Panagis, E. Sakkopoulos, G. Tzimas, A. Tsakalidis

**April 2006 Proceedings of the 2006 ACM symposium on Applied computing SAC '06****Publisher:** ACM PressFull text available: [pdf\(356.17 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this work we present techniques and tools that enable effective reverse engineering procedures for web applications that were developed using the promising ASP.NET technology. We deal with model-driven development in its reverse aspect by implementing reverse engineering methods. Our implemented methods model web applications using a well-known, web oriented and robust language, namely WebML. This is, to the authors' best knowledge, a novel re-engineering transformation. In this paper we prop ...

**Keywords:** ASP.NET, WebML, reverse engineering, web applications**18 Reverse engineering: Search based reverse engineering**

Brian S. Mitchell, Spiros Mancoridis, Martin Traverso

**July 2002 Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02****Publisher:** ACM PressFull text available: [pdf\(208.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

In this paper we describe a two step process for reverse engineering the software architecture of a system directly from its source code. The first step involves clustering the modules from the source code into abstract structures called subsystems. The second step involves reverse engineering the subsystem-level relations using a formal (and visual) architectural constraint language. We use search techniques to accomplish both of these steps, and have implemented a suite of integrated tools to ...

**19 A collaborative demonstration of reverse engineering tools**

Margaret-Anne D. Storey, Susan Elliott Sim, Kenny Wong

**April 2002 ACM SIGAPP Applied Computing Review**, Volume 10 Issue 1**Publisher:** ACM PressFull text available: [pdf\(647.46 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a *collaborative structured demonstration* of reverse engineering tools that was presented at a working session at WCRE 2001 in Stuttgart, Germany. A *structured demonstration* is a hybrid tool evaluation technique that combines elements from experiments, case studies, technology demonstrations, and benchmarking. The essence of the technique is to facilitate learning about software engineering tools using a common set of tasks. The collaborative experience discusse ...

**Keywords:** benchmark, empirical study, program comprehension, reengineering, research methodology, reverse engineering, tool evaluation**20****Generic fuzzy reasoning nets as a basis for reverse engineering relational database**

 applications

Jens H. Jahnke, Wilhelm Schäfer, Albert Zündorf

November 1997 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th European conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering ESEC '97/FSE-5**, Volume 22 Issue 6

Publisher: Springer-Verlag New York, Inc., ACM Press

Full text available:  [pdf\(1.45 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

 **PORTAL**  
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login  
 Search:  The ACM Digital Library  The Guide

## THE ACM DIGITAL LIBRARY

 Feedback Report a problem Satisfaction survey

Terms used reverse engineering

Found 61,162 of 184,245

Sort results by   Save results to a Binder  
 Search Tips  Open results in a new window

Display results

Try an [Advanced Search](#)  
 Try this search in [The ACM Guide](#)

Results 21 - 40 of 200 Result page: [previous](#) **1** [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)  
 Best 200 shown

Relevance scale 

**21** [Finding approximate shape regularities in reverse engineered solid models bounded by simple surfaces](#) 

 Frank Langbein, Bruce I. Mills, A. Dave Marshall, Ralph R. Martin  
 May 2001 **Proceedings of the sixth ACM symposium on Solid modeling and applications**

Publisher: ACM Press

Full text available:  pdf(1.03 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Current reverse engineering systems are able to generate simple valid boundary representation (B-rep) models from 3D range data. Such models suffer from various inaccuracies caused by noise in the input data and algorithms. The quality of reverse engineered geometric models can potentially be improved by finding candidate shape regularities in such an initial model, and imposing a suitable subset of them on the model by using constraints, in a postprocessing step called *beautification*. ...

**Keywords:** beautification, geometric interrogations and reasoning, reverse engineering, shape regularities, similarity

**22** [Insight: reverse engineer case tool](#) 

Norman Rajala, Djenana Campara, Nikolai Mansurov  
 May 1999 **Proceedings of the 21st international conference on Software engineering**

Publisher: IEEE Computer Society Press

Full text available:  pdf(530.84 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

**Keywords:** reverse engineer, software maintenance

**23** [A visual requirements validation environment for the reverse engineering of formal specifications from rapid prototypes](#) 

 M. B. Ozcan, I. Morrey  
 December 1995 **ACM SIGSOFT Software Engineering Notes**, Volume 20 Issue 5

Publisher: ACM Press

Full text available:  pdf(596.43 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper describes a research project whose aim is the use of requirements visualisation techniques in the construction of an environment for the reverse engineering of validated formal specifications from rapid prototypes. The work will build on established research by the proposers in the animation of model-based and algebraic formal specifications. The report examines the current problems with requirements engineering and looks at solutions based on software prototyping and executable formal ...

## 24 Reverse engineering framework reuse interfaces

 Jukka Viljamaa

September 2003 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-11**, Volume 28 Issue 5

Publisher: ACM Press

Full text available:  pdf(313.32 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

*Object-oriented application frameworks* provide an established way of reusing the design and implementation of applications in a specific domain. Using a framework for creating applications is not a trivial task, however, and special tools are needed for supporting the process. Tool support, in turn, requires explicit specification of the *reuse interfaces* of frameworks. Unfortunately these specifications typically become quite extensive and complex for non-trivial frameworks. In thi ...

**Keywords:** documentation, formal concept analysis, framework, pattern, reuse, reverse engineering

## 25 Basic concepts for an HDL reverse engineering tool-set

Gunther Lehmann, Bernhard Wunder, Klaus D. Müller-Glaser

January 1997 **Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design**

Publisher: IEEE Computer Society

Full text available:  pdf(298.21 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

 Publisher Site

Designer's productivity has become the key-factor of the development of electronic systems. An increasing application of design data reuse is widely recognized as a promising technique to master future design complexities. Since the intellectual property of a design is more and more kept in software-like hardware description languages (HDL), successful reuse depends on the availability of suitable HDL reverse engineering tools. This paper introduces new concepts for an integrated HDL reverse eng ...

**Keywords:** VHDL Verilog Hardware Description Reuse Reverse Engineering Hypertext CASE Visualization Productivity Design Process Analysis Control Flow ADA Graphical Symbol

## 26 Static control-flow analysis for reverse engineering of UML sequence diagrams

 Atanas Rountev, Olga Volgin, Miriam Reddoch

September 2005 **ACM SIGSOFT Software Engineering Notes , The 6th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering PASTE '05**, Volume 31 Issue 1

Publisher: ACM Press

Full text available:  pdf(220.28 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

UML sequence diagrams are commonly used to represent the interactions among collaborating objects. Reverse-engineered sequence diagrams are constructed from

existing code, and have a variety of uses in software development, maintenance, and testing. In static analysis for such reverse engineering, an open question is how to represent the intraprocedural flow of control from the code using the control-flow primitives of UML 2.0. We propose simple UML extensions that are necessary to capture gener ...

27 Leveraging IBM visual age for C++ for reverse engineering tasks

Johannes Martin

November 1999 **Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available:  pdf(283.87 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The IBM VisualAge for C++ development environment provides tool writers with the ability to query the compiler's internal data structures for information on the programs being compiled. This paper shows how these features can be used to write data extractors for supplying data to common reverse engineering tools, while significantly reducing the complexity of the data extractors and the time needed to develop these as compared to using traditional approaches and programming tools.

28 Dynamic analysis for reverse engineering and program understanding

 Eleni Stroulia, Tarja Systä

April 2002 **ACM SIGAPP Applied Computing Review**, Volume 10 Issue 1

Publisher: ACM Press

Full text available:  pdf(150.21 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The main focus of program understanding and reverse engineering research has been on modeling the structure of a program by examining its code. This has been the result of the nature of the systems investigated and the perceived goals of the reverse engineering activities. The types of systems under investigation have changed, however, and the maintenance objectives have evolved. Many legacy systems today are object-oriented and component-based. One of the most prominent maintenance objectives i ...

**Keywords:** dynamic analysis, reverse engineering

29 Visualization techniques II: An open toolkit for prototyping reverse engineering visualizations

Alexandru Telea, Alessandro Maccari, Claudio Riva

May 2002 **Proceedings of the symposium on Data Visualisation 2002 VISSYM '02**

Publisher: Eurographics Association

Full text available:  pdf(480.08 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Maintenance and evolution of complex software systems (such as mobile telephones) involves activities such as reverse engineering (RE) and software visualization. Although several RE tools exist, we found their architecture hard to adapt to the domain and problem specific requirements posed by our current practice in Nokia. In this paper, we present an open architecture which allows easy prototyping of RE data exploration and visualization scenarios for a large range of domain models. We pay spe ...

30 Reverse engineering of legacy systems: a path toward success

 Alex Quilici

April 1995 **Proceedings of the 17th international conference on Software engineering**

Publisher: ACM Press

Full text available: [pdf\(395.51 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

31 [Automatic instruction scheduler retargeting by reverse-engineering](#)

 Matthew J. Bridges, Neil Vachharajani, Guilherme Ottoni, David I. August  
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6

Publisher: ACM Press

Full text available: [pdf\(188.13 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In order to generate high-quality code for modern processors, a compiler must aggressively schedule instructions, maximizing resource utilization for execution efficiency. For a compiler to produce such code, it must avoid structural hazards by being aware of the processor's available resources and of how these resources are utilized by each instruction. Unfortunately, the most prevalent approach to constructing such a scheduler, manually discovering and specifying this information, is both tedi ...

**Keywords:** automatic retargeting, compilers, instruction scheduling, reverse-engineering, structural hazard

32 [Approximate symmetry detection for reverse engineering](#)

 B. I. Mills, F. C. Langbein, A. D. Marshall, R. R. Martin  
May 2001 **Proceedings of the sixth ACM symposium on Solid modeling and applications**

Publisher: ACM Press

Full text available: [pdf\(751.33 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The authors are developing an automated reverse engineering system for reconstructing the shape of simple mechanical parts. B-rep models are created by fitting surfaces to point clouds obtained by scanning an object using a 3D laser scanner. The resulting models, although valid, are often not suitable for purposes such as redesign because expected regularities and constraints are not present. This information is lost because each face of the model is determined independently. A global approac ...

**Keywords:** approximate symmetry, beautification, geometric interrogations and reasoning, reverse engineering

33 [Software evolution: Object naming analysis for reverse-engineered sequence diagrams](#)

 Atanas Rountev, Beth Harness Connell  
May 2005 **Proceedings of the 27th international conference on Software engineering**

Publisher: ACM Press

Full text available: [pdf\(230.57 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

UML sequence diagrams are commonly used to represent object interactions in software systems. This work considers the problem of extracting UML sequence diagrams from existing code for the purposes of software understanding and testing. A static analysis for such reverse engineering needs to map the interacting objects from the code to sequence diagram objects. We propose an interprocedural dataflow analysis algorithm that determines precisely which objects are the receivers of certain messages, ...

**Keywords:** UML, reverse engineering, static analysis

- 34 Workshop and conference summaries: Practical data exchange for reverse engineering frameworks: some requirements, some experience, some headaches

 Michael W. Godfrey

January 2001 **ACM SIGSOFT Software Engineering Notes**, Volume 26 Issue 1

Publisher: ACM Press

Full text available:  pdf(329.81 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Reverse engineering systems hold great promise in aiding developers regain control over long-lived software projects whose architecture has been allowed to "drift". However, it is well known that these systems have relative strengths and weaknesses, and to date relatively little work has been done on integrating various subtools within other reverse engineering systems. The design of a common interchange format for data used by reverse engineering tools is therefore of critical importance. In thi ...

**Keywords:** TAXFORM, data exchange, fact extractors, reverse engineering

- 35 Reverse engineering and reengineering of a large serial system into a distributed-parallel version

 Kosmas Karadimitriou, John Tyler, N. E. Brener  
February 1995 **Proceedings of the 1995 ACM symposium on Applied computing**

Publisher: ACM Press

Full text available:  pdf(867.75 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

**Keywords:** BNDFKG, PVM, distributed computing, reengineering, reverse engineering

- 36 Reverse engineering Ada into HOOD

 R. H. Pierce, S. D. Bluck  
December 1990 **Proceedings of the conference on TRI-ADA '90**

Publisher: ACM Press

Full text available:  pdf(570.52 KB) Additional Information: [full citation](#), [references](#)

- 37 A reverse engineering tool for precise class diagrams

Yann-Gaël Guéhéneuc  
October 2004 **Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available:  pdf(333.40 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Developers use class diagrams to describe the architecture of their programs intensively. Class diagrams represent the structure and global behaviour of programs. They show the programs classes and interfaces and their relationships of inheritance, instantiation, use, association, aggregation and composition. Class diagrams could provide useful data during programs maintenance. However, they often are obsolete and imprecise: They do not reflect the real implementation and behaviour of program ...

- 38 A cognitive and user centric based approach for reverse engineering tool design

Iyad Zayour, Timothy C. Lethbridge

November 2000 **Proceedings of the 2000 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available: [pdf\(69.23 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Reverse engineering tools aimed at facilitating software maintenance suffer from low adoption. Many are developed, but few are used by software engineers in performing their maintenance work. We introduce an approach for tool design that is aimed at increasing the adoptability potential of tools. Our approach is based on applying cognitive analysis to identify cognitively difficult aspects of maintenance work, then deriving cognitive requirements to address these difficulties. The approach is des ...

39 [PRISM: a reverse engineering toolset](#)

 Lonnie R. Welch

November 1997 **ACM SIGAda Ada Letters**, Volume XVII Issue 6

Publisher: ACM Press

Full text available: [pdf\(440.76 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper presents a process for the reengineering of computer-based control systems, and describes tools that automate portions of the process. The intermediate representation (IR) for capturing features of computer-based systems during reverse engineering is presented. A novel feature of the IR is that it incorporates the control system software architecture, a view that enables information to be captured at five levels of granularity: the program level, the task level, the package level, the ...

40 [Tutorials: Reverse engineering of object oriented code](#)

 Paolo Tonella

May 2005 **Proceedings of the 27th international conference on Software engineering**

Publisher: ACM Press

Full text available: [pdf\(57.01 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

During software evolution, programmers devote most of their effort to the understanding of the structure and behavior of the system. For Object-Oriented code, this might be particularly hard, when multiple, scattered objects contribute to the same function. Design views offer an invaluable help, but they are often not aligned with the code, when they are not missing at all. This tutorial describes some of the most advanced techniques that can be employed to reverse engineer several design views f ...

**Keywords:** diagram recovery, object oriented programming, static code analysis

Results 21 - 40 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)